



- (51) **International Patent Classification:**
G06F 9/30 (2006.01) *G06F 9/38* (2006.01)
- (21) **International Application Number:**
PCT/GB2016/051001
- (22) **International Filing Date:**
11 April 2016 (11.04.2016)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
1509743.9 5 June 2015 (05.06.2015) GB
- (71) **Applicant:** ARM LIMITED [GB/GB]; 110 Fulbourn Road, Cherry Hinton, Cambridge Cambridgeshire CB1 9NJ (GB).
- (72) **Inventors:** PENTON, Anthony John; c/o ARM Limited, 110 Fulbourn Road, Cherry Hinton, Cambridge Cambridgeshire CB1 9NJ (GB). CRASKE, Simon John; c/o ARM Limited, 110 Fulbourn Road, Cherry Hinton, Cambridge Cambridgeshire CB1 9NJ (GB). VASEKIN, Vladimir; c/o ARM Limited, 110 Fulbourn Road, Cherry Hinton, Cambridge Cambridgeshire CB1 9NJ (GB).
- (74) **Agent:** ROBINSON, Nigel; D Young & Co LLP, 120 Holborn, London EC1N 2DY (GB).

(81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

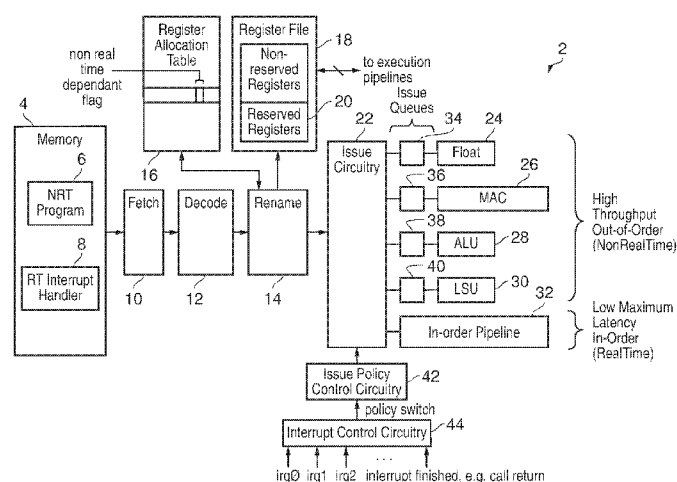
(54) **Title:** INSTRUCTION ISSUE ACCORDING TO IN-ORDER OR OUT-OF-ORDER EXECUTION MODES

FIG. 1

(57) **Abstract:** Apparatus for processing data (2) includes issue circuitry (22) for issuing program instructions (processing operations) to execute either within real time execution circuitry (32) or non real time execution circuitry (24, 26, 28, 30). Registers within a register file (18) are marked as non real time dependent registers if they are allocated to store a data value which is to be written by an uncompleted program instruction issued to the non real time execution circuitry and not yet completed. Issue policy control circuitry (42) responds to a trigger event to enter a real time issue policy mode to control the issue circuitry (22) to issue candidate processing operations (such as program instruction, micro-operations, architecturally triggered processing operations etc.) to one of the non real time execution circuitry or the real time execution circuitry in dependence upon whether that candidate processing operation reads a register marked as a non real time dependent register.



INSTRUCTION ISSUE ACCORDING TO IN-ORDER OR OUT-OF-ORDER EXECUTION MODES

This disclosure relates to data processing systems. More particularly, this disclosure relates to issue policy control within data processing systems.

5 It is known to provide data processing systems in which program instructions may be issued to pipelines under control of an issue policy.

Viewed from one aspect the present invention provides apparatus for processing data comprising:

- 10 issue circuitry to issue processing operations for execution;
non real time execution circuitry to execute processing operations issued to said non real time execution circuitry by said issue circuitry;
real time execution circuitry to execute processing operations issued to said real time execution circuitry by said issue circuitry;
- 15 one or more registers to store data values, respective registers of said one or more registers having associated indicators to indicate for a given register when said given register is a non real time dependent register allocated to store a data value written by an uncompleted processing operation issued to said non real time execution circuitry; and
- 20 issue policy control circuitry to respond to a trigger event to enter a real time mode to control said issue circuitry to issue a candidate processing operation to one of said non real time execution circuitry and said real time execution circuitry in dependence upon whether that said candidate processing operation reads a non real time dependent register.

25 Viewed from another aspect the present disclosure provides apparatus for processing data comprising:

- issue means for issuing processing operations for execution;
non real time execution means for executing processing operations issued to said non real time execution means by said issue means;
real time execution means for executing processing operations issued to said real time execution means by said issue means;
- 30 one or more register means for storing data values, respective register means of said one or more register means having associated indicators to indicate for a given register means when said given register means is a non real time dependent register means allocated to store a data value written by an uncompleted processing operation issued to said non real time execution means; and
- 35 issue policy control means for responding to a trigger event to enter a real time mode to control said issue circuitry to issue a candidate processing operation to one of said

non real time execution means and said real time execution means in dependence upon whether that said candidate processing operation reads a non real time dependent register means.

5 Viewed from a further aspect the present invention provides a method of processing data comprising:

issuing processing operations for execution to a selected one of non real time execution circuitry and real time execution circuitry;

10 storing data values in one or more registers, respective registers of said one or more registers having associated indicators to indicate for a given register when said given register is a non real time dependent register allocated to store a data value written by an uncompleted processing operation issued to said non real time execution circuitry; and

15 in response to a trigger event, entering a real time mode to control said issue of a candidate processing operation to one of said non real time execution circuitry and said real time execution circuitry in dependence upon whether that said candidate processing operation reads a non real time dependent register.

Example embodiments will now be described, by way of example only, with reference to the accompanying drawings in which:

20

Figure 1 schematically illustrates a data process apparatus including non-real time execution circuitry and real time execution circuitry;

Figure 2 schematically illustrates a program instruction including a destination register specifier;

25 Figure 3 is a flow diagram schematically illustrating switching between a real time issue policy and a non real time issue policy;

Figure 4 schematically illustrates a real time issue policy;

Figure 5 is a flow diagram schematically illustrating marking registers as non real time dependent;

30 Figure 6 is a flow diagram schematically illustrating the clearing of non real time dependent marking for registers; and

Figure 7 schematically illustrates example code including an interrupt handler and to which of the non real time execution circuitry and the real time execution circuitry respective instructions are issued.

35

Figure 1 schematically illustrates an apparatus for data processing 2 including a memory 4 storing a non real time program 6 (such as, for example, a non critical program

permitting a user to configure a display) and a real time program 8 (such as, for example, a real time interrupt handler for performing car engine management or antilock braking system operation). Program instructions read from the memory 4 are fetched by a fetch stage 10 (fetch circuitry) and then passed in turn to a decode stage 12 (decode circuitry) and a register renaming stage 14 (register renaming circuitry). The data processing apparatus 2 is able to operate in a non real time execution mode in which out-of-order pipelines 24, 26, 28, 30 are used to support out-of-order processing. The data processing apparatus 2 also operates in a real time mode in which an in-order pipeline 32 is used to support in-order processing. The out-of-order pipelines 24, 26, 28, 30 are, in this example embodiment, the non real time execution circuitry that typically has a higher average instruction execution rate than real time execution circuitry in the form of the in-order pipeline 32. The real time execution circuitry typically has a lower maximum instruction execution latency than the non real time execution circuitry and also has a lower variation in instruction execution latency than the non real time execution circuitry. In general terms, program instructions processed by the in-order pipeline 32 will typically execute a slower rate, but with more deterministic behavior.

When operating in the non real time mode, the rename stage 14 performs register renaming upon the decoded program instructions using a register allocation table 16 to store register mapping data for the registers within the register file 18. A portion of the registers within the register file 18 are reserved registers 20 which are reserved for use by the real time execution circuitry are not utilized by the non real time execution circuitry.

Program instructions (i.e. one or more processing operations – a program instruction may be broken down into multiple processing operations, which could be executed by the same or different pipelines from the rename stage 14 (whether renaming has or has not been performed) are passed to issue circuitry 22 where they are issued (via respective issue queues 34, 36, 38, 40) to either the non real time execution circuitry comprising execution pipelines 24, 26, 28, 30 in accordance with out-of-order processing techniques, or alternatively to the in-order pipeline 32 in accordance with in-order processing techniques. Issue policy control circuitry 42 serves to control (select) the issue policy that is used. The issue policy may be either a non real time issue policy or a real time issue policy. Interrupt control circuitry 44 receives external interrupt signals and triggers execution of associated interrupt handling programs, together with triggering a policy switch, where specified by the interrupt configuration data, from non real time processing to real time processing.

It may be desirable for some program code to be executed in accordance with the real time issue policy in order that it may have more deterministic behaviour (e.g. a lower maximum latency) even though the non real time issue policy may be able to achieve higher average instruction throughput. In the present example embodiment switching from non real time issue policy to real time issue policy may be controlled by a trigger event in the form of a received interrupt signal. Other forms of trigger event are also possible, such as a software interrupt (SWI) or execution of an explicit switch instruction should this be provided. A reset event may be used to switch back from the real time issue policy to the non real time issue policy. The reset event again can take a number of different forms, such as completion of execution of interrupt handling program instructions, a call return, an exception return, or execution of a specific switch instruction within the program instruction stream.

When a switch is made from the non real time issue policy to the real time issue policy, there will typically be a number of pending instruction "in flight" (issued, partially processed, but not yet completed (retired), i.e. uncompleted processing operations) within the out-of-order pipelines 24, 26, 28, 30. Some of these instructions may write data values to destination registers within the register file 18. Given the non deterministic behaviour of the out-of-order pipelines 24, 26, 28, 30, it may be a relatively long time before such output operands are written to the destination registers concerned for these "in flight" instructions within the non real time execution circuitry. As an example, a load to a register within the register file 18 performed by the load store unit 30 may result in a cache miss and then a long latency main memory access requiring many hundreds of processing cycles before that destination register is written with the data value returned from main memory.

The issue circuitry 22 is able to reduce the impact of this delay, at least to some extent, by using a real time issue policy in which non real time dependent registers are identified in the register allocation table 16 and an instruction which reads such a non real time dependent register is issued by the issue circuitry 22 to the non real time execution circuitry rather than to the real time execution circuitry. Thus, the non real time execution circuitry is able to use its existing dependency control mechanisms to ensure that the program instruction which reads the non real time dependent register is executed in an appropriate order relative to writing of that non real time dependent register. However, using the real time issue policy, program instructions which do not read any non real time dependent registers are issued into the real time execution circuitry in the form of the in-order pipeline 32. This permits processing which is not dependent upon pending

processing within the non real time execution circuitry to proceed within the real time execution circuitry without undue delay.

The identification of registers within the register file 18 as non real time dependent registers may be achieved using a non real time dependent flag within the data held by the register allocation table. When the rename stage 14 renames a register being used by the non real time execution circuitry and notes that a write is to be made to that register by the non real time execution circuitry, then the non real time dependent flag is set for that register. The out-of-order processing techniques utilized by the non real time execution circuitry may include techniques such as speculative writing of data values to registers within the register file 18. In this case, a program instruction executed by the non real time execution circuitry will be completed when its corresponding program instruction is retired indicating that it is no longer speculative and the data value (which may have been generated some time before) held within the non real time dependent register is confirmed as architectural state of the system.

In the above discussion, the processing workload has been considered to be formed of program instructions which are passed either to the non real time execution circuitry or the real time execution circuitry. It will be appreciated by those in this field that the processing workload associated with program instructions may in practice, for example, be broken down into smaller units, such as breaking down a program instruction into microoperations, or processing operations associated with changes of state of the system as a whole, e.g. processing operations associated with interrupt signals which may trigger state saving operations, status register writes and the like which are performed in addition to and alongside processing operations specified by program instructions. Such processing operations may be issued into the non real time execution circuitry or the real time execution circuitry by the issue circuitry 22 in dependence upon whether the non real time issue policy or the real time issue policy is in force and in accordance with the techniques discussed above in relation to the real time issue policy and its relationship to non real time dependent registers.

Figure 2 schematically illustrates a program instruction 46 which includes an opcode 48, a first source register specifier 50, a second source register specifier 52 and a destination register specifier 54. If the program instruction 46 is issued into the non real time execution circuitry, e.g. in the form of the out-of-order pipelines 24, 26, 28, 40, then the register within the register file 18 corresponding to the destination register specifier 54 is marked within the register allocation table 16 by a flag to indicate that it is a non real

time dependent register. This flag may in turn may be used when the real time issue policy is in force to ensure that a program instruction (or more generally processing operations as discussed above) that reads that non real time dependent register before its non real time dependence flag status has been cleared will be issued into the non real time execution circuitry and so be properly ordered with respect to the preceding instruction which writes that non real time dependent register.

Figure 3 is a flow diagram schematically illustrating switching between a non real time issue policy and a real time issue policy. Processing starts at step 56 with the non real time issue policy in use. Processing waits at step 56 until a trigger event is detected. This trigger event may be, for example, a received physical interrupt signal, a software interrupt or execution of an explicit switch instruction. Other forms of trigger event are also possible. When a trigger event has been detected, then step 58 serves to flush the program instructions (or processing operations) held within the issue queues 34, 36, 38, 40. (in some embodiments one or more of the issue queues may not be flushed, e.g. instructions certain to commit can be left, whether or not instructions must be flushed can depend upon, for example, how the out-of-order pipeline resolves branches and/or faults). As the instructions within the issue queues 34, 36, 38, 40 are flushed before their execution starts within the out-of-order pipelines 24, 26, 28, 30, there are no uncompleted "in flight" processing operation which will write to a non real time dependent register for those flushed instructions and accordingly the non real time dependent flags for those flushed instructions can be cleared within the register allocation table 16. Step 60 switches to the real time issue policy as discussed above. Processing waits at step 62 until a reset event is detected. The reset event could take a variety of different forms, such as, for example, an exception return or an explicit reset instruction. When a reset event is detected, then step 64 switches from the real time issue policy to the non real time issue policy.

Figure 4 is a flow diagram schematically illustrating operation in accordance with the real time issue policy. Processing waits at step 66 until there is a candidate instruction (processing operation) to issue. Step 68 then determines whether the candidate instruction reads a register which is marked as a non real time dependent register within the register allocation table 16. If the instruction does not read any non real time dependent registers, then it is issued to the real time execution circuitry at step 70. If the instruction does read a register marked as a non real time dependent register, then the instruction is issued to the non real time execution circuitry at step 72.

In some example embodiments there may be an additional step between steps 68 and 70 which serves to issue at least some instructions to the non real time execution circuitry during the real time mode even if those instructions do not read any registers marked as non real time dependent. Thus, in such embodiments not all instructions that do not read non real time dependent registers need to be sent to the real time execution circuitry. Instead a proportion may continue to be sent to the non real time execution circuitry such that a desired balance between throughput and latency may be achieved. The proportion of instructions “voluntarily” sent to the non real time execution circuitry during the real time mode could be varied (e.g. user configured) depending on the throughput vs latency balance desired for a particular implementation or use.

In some example embodiments a user defined tolerated-latency-setting may be provided (e.g. a flag(s) in configuration registers) that corresponds to different sets of processing operations continuing to be issued by the issue circuitry to the non real time execution circuitry even though they do not read any non real time dependent registers. The candidate processing operations to be issued will have different associated maximum latency value when executed by the non real time processing circuitry, e.g. in relative order of maximum latency – simple arithmetic and logic processing operations (ADD, SUB, AND, OR, ...), complex arithmetic processing operations (DIV, SQRT), flushable load/store processing operations, and non-flushable load/store processing operations (e.g. access to memory regions marked strongly ordered). Different sets of processing operation selected in accordance with this ordering may continue to be issued to the non real time execution circuitry depending on the user setting indicating the maximum tolerated latency required by the user.

Figure 5 is a flow diagram schematically illustrating marking of registers as non real time dependent registers. At step 74 processing waits until an instruction is encountered by the rename stage 14 that is to be issued by the issue circuitry 22 to the non real time execution circuitry (i.e. is subject to the non real time issue policy). When an instruction is to be issued to the non real time execution circuitry, step 76 determines whether it writes to a register (a destination register). If the instruction does write to a register, then step 78 marks that destination register as a non real time dependent register.

Figure 6 is a flow diagram schematically illustrating one mechanism for clearing a non real time dependent register flag. At step 80 processing waits until an instruction is ready to retire from the non real time execution circuitry (e.g. it has completed execution and become non speculative). When an instruction retired, any destination registers

associated with that instruction are henceforth treated as architectural state (non speculative state) and the non real time dependent flags associated with any of those destination registers are cleared. These operations occur at step 82. As previously mentioned, non real time dependent flags for destination registers may also be cleared for
5 program instructions (processing operations) flushed from the issue queues 34, 36, 38, 40 before they are issued into the out-of-order pipelines 24, 26, 28, 30 respectively. Other flag clearing mechanisms are also possible.

Figure 7 schematically illustrates example program code including non real time
10 code 84 whose execution is interrupted by an interrupt signal triggering execution of real time code 86. The real time code 86 is an interrupt handler. The non real time code 84 is interrupted after a multiply instruction MUL has been issued into the multiply accumulate pipeline 88 of non real time execution circuitry. The multiply instruction MUL is accordingly "in flight" and its destination register rC is marked a non real time dependent
15 register. The instructions of the non real time code 84 following the MUL instruction, namely the STR instruction and the LDR instruction, are flushed before they commence execution within the out-of-order pipelines.

At the beginning of the real time code a stack writing instruction PUSH is to be
20 performed in respect of register r0, which corresponds to the register rC, i.e. is a non real time dependent register as it is to be written by the "in flight" MUL instruction within the multiply accumulate pipeline 88. Accordingly, even though the interrupt which has arisen has switched the system to use of the real time issue policy, the stack writing instruction PUSH is issued into the non real time execution circuitry. Conversely, the instructions
25 within the real time code 86 following the stack writing instruction PUSH do not make any reads of non real time dependent registers. Accordingly, they are issued into the in-order pipeline 90 by the issue circuitry 92. The in-order pipeline circuitry 90 uses reserved registers 94 which are available for its immediate use. These reserved registers 94 may include registers which have an architectural specifier corresponding to a non real time
30 dependent register as the real time issue policy will ensure that they are only issued into the in-order pipeline 90 if they do not read from the non real time dependent register concerned.

Although illustrative embodiments have been described in detail herein with
35 reference to the accompanying drawings, it is to be understood that the claims are not limited to those precise embodiments, and that various changes, additions and modifications can be effected therein by one skilled in the art without departing from the

scope and spirit of the appended claims. For example, various combinations of the features of the dependent claims could be made with the features of the independent claims.

CLAIMS

1. Apparatus for processing data comprising:
issue circuitry to issue processing operations for execution;
5 non real time execution circuitry to execute processing operations issued to said non real time execution circuitry by said issue circuitry;
real time execution circuitry to execute processing operations issued to said real time execution circuitry by said issue circuitry;
one or more registers to store data values, respective registers of said one or more
10 registers having associated indicators to indicate for a given register when said given register is a non real time dependent register allocated to store a data value written by an uncompleted processing operation issued to said non real time execution circuitry; and
issue policy control circuitry to respond to a trigger event to enter a real time mode to control said issue circuitry to issue a candidate processing operation to one of said non
15 real time execution circuitry and said real time execution circuitry in dependence upon whether that said candidate processing operation reads a non real time dependent register.
2. Apparatus as claimed in claim 1, wherein when in a non real time mode said issue policy control circuitry controls said issue circuitry to issue processing operations to said
20 non real time execution circuitry.
3. Apparatus as claimed in claim 1, wherein when said candidate processing operation reads a non real time dependent register, said candidate processing operation is issued to
25 said non real time execution circuitry.
4. Apparatus as claimed in any one of claims 1, 2 and 3, wherein when said candidate processing operation is independent of any a non real time dependent register, said candidate processing operation is issued to said real time execution circuitry.
- 30 5. Apparatus as claimed in any one of claims 1, 2 and 3, when in said real time mode said issue circuitry issues at least some processing operations that do not read a non real time dependent register to said non real time execution circuitry.
6. Apparatus as claimed in claim 5, wherein, in said real time mode, a user defined
35 tolerated-latency-setting controls said issue circuitry to issue to said non real time execution circuitry processing operations that do not read a non real time dependent register and have less than a tolerated maximum interrupt latency when executed by said

non real time execution circuitry.

7. Apparatus as claimed in claim 6, wherein said user defined tolerated-latency-setting corresponds to one of a plurality of sets of processing operations having less than said
5 tolerated maximum interrupt latency.

8. Apparatus as claimed in claim 7, wherein said processing operations include in relative order of maximum interrupt latency when executed by said non real time execution circuitry: simple arithmetic and logic processing operations, complex arithmetic processing
10 operations, flushable load and store operations, and non-flushable load and store operations.

9. Apparatus as claimed in any one of the preceding claims, wherein said candidate processing operation corresponds to a candidate program instruction.
15

10. Apparatus as claimed in any one of the preceding claims, wherein said non real time execution circuitry is out-of-order execution circuitry to perform out-of-order execution of said processing operations.

20 11. Apparatus as claimed in claim 10, wherein said out-of-order execution circuitry comprises register renaming circuitry to support speculative writing of data values to said one or more registers.

25 12. Apparatus as claimed in claim 11, wherein program instruction completion includes retirement of a speculatively executed program instruction.

13. Apparatus as claimed in any one of the preceding claims, wherein said real time execution circuitry is in-order execution circuitry to perform in-order execution of said processing operations.
30

14. Apparatus as claimed in any one of the preceding claims, wherein said trigger event is an interrupt trigger event triggering a redirection of program flow to execute interrupt handling processing operations.

35 15. Apparatus as claimed in claim 14, comprising interrupt control circuitry to receive one or more interrupt signals, wherein said interrupt trigger event corresponds to receipt of an interrupt signal by said interrupt control circuitry.

16. Apparatus as claimed in any of the preceding claims, wherein said issue circuitry comprises an issue queue to store processing operations awaiting issue and in response to said trigger event, said issue circuitry flushes said issue queue.

5

17. Apparatus as claimed in any one of the preceding claims, wherein said associated indicators comprise respective flag values for said one or more registers, said flag values marking upon issue of a processing operation to said non real time execution circuitry registers that are output operand destination registers of said processing operations.

10

18. Apparatus as claimed in any one of the preceding claims comprising one or more reserved registers reserved for use to store respective operands value of processing operations executed by said real time execution circuitry.

15

19. Apparatus as claimed in any one of the preceding claims, wherein said issue policy control circuitry responds to a reset event to enter a non real time mode to control said issue circuitry to issue said candidate processing operation to said non real time execution circuitry.

20

20. Apparatus as claimed in claim 14 and claim 19, wherein said reset event is completion of execution of said interrupt handling program instructions.

21. Apparatus as claimed in any one of the preceding claims, wherein said non real time execution circuitry has a higher average instruction execution rate than said real time execution circuitry.

25

22. Apparatus as claimed in any one of the preceding claims, wherein real time execution circuitry has a lower maximum instruction execution latency than said non real time execution circuitry.

30

23. Apparatus as claimed in any one of the preceding claims, wherein real time execution circuitry has a lower variation in instruction execution latency than said non real time execution circuitry.

35

24. Apparatus for processing data comprising:
issue means for issuing processing operations for execution;
non real time execution means for executing processing operations issued to said

non real time execution means by said issue means;

real time execution means for executing processing operations issued to said real time execution means by said issue means;

one or more register means for storing data values, respective register means of said
5 one or more register means having associated indicators to indicate for a given register means when said given register means is a non real time dependent register means allocated to store a data value written by an uncompleted processing operation issued to said non real time execution means; and

issue policy control means for responding to a trigger event to enter a real time
10 mode to control said issue circuitry to issue a candidate processing operation to one of said non real time execution means and said real time execution means in dependence upon whether that said candidate processing operation reads a non real time dependent register means.

15 25. A method of processing data comprising:

issuing processing operations for execution to a selected one of non real time execution circuitry and real time execution circuitry;

storing data values in one or more registers, respective registers of said one or more registers having associated indicators to indicate for a given register when said given
20 register is a non real time dependent register allocated to store a data value written by an uncompleted processing operation issued to said non real time execution circuitry; and

in response to a trigger event, entering a real time mode to control said issue of a candidate processing operation to one of said non real time execution circuitry and said real time execution circuitry in dependence upon whether that said candidate processing
25 operation reads a non real time dependent register.

1/5

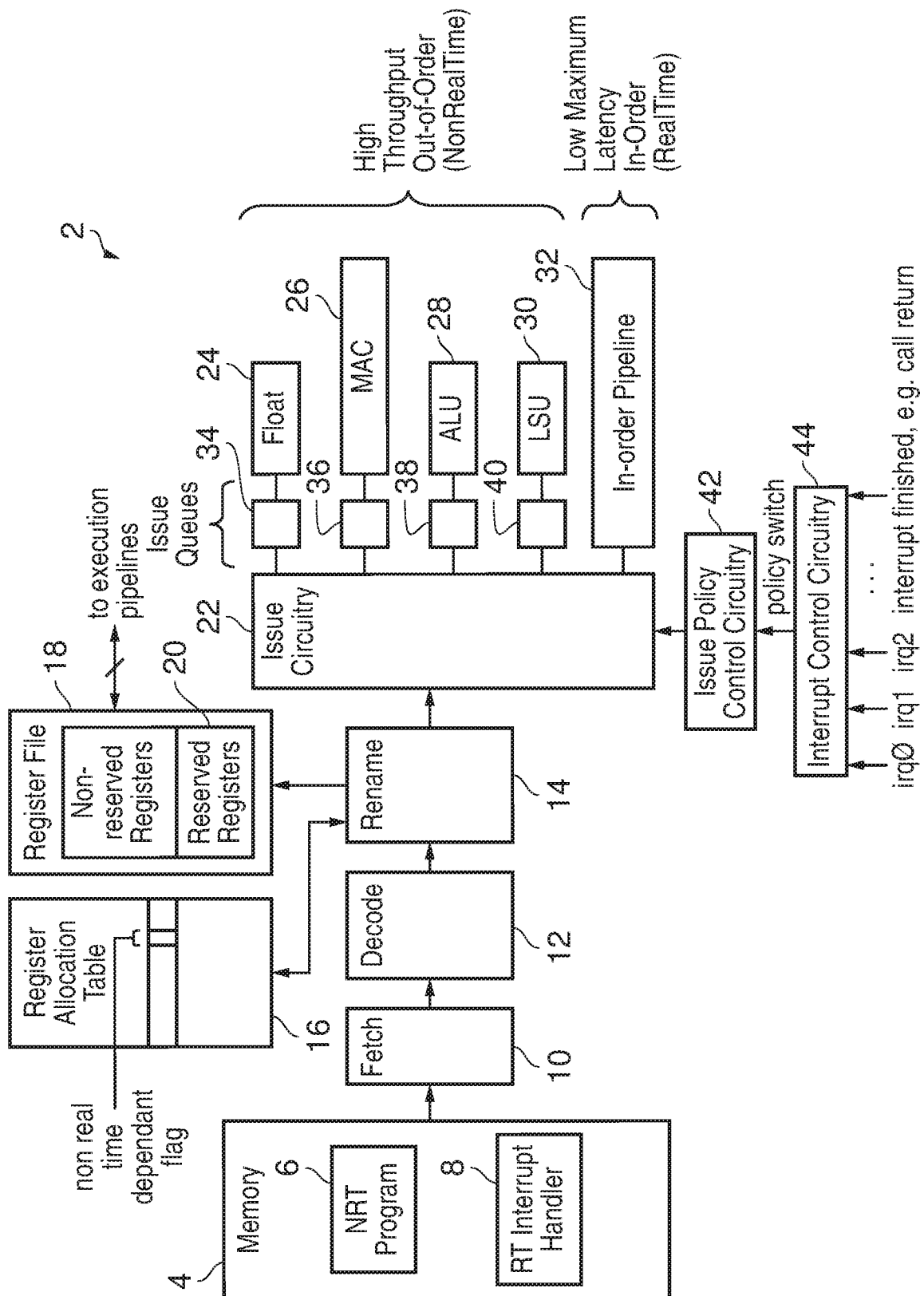


FIG. 1

2/5

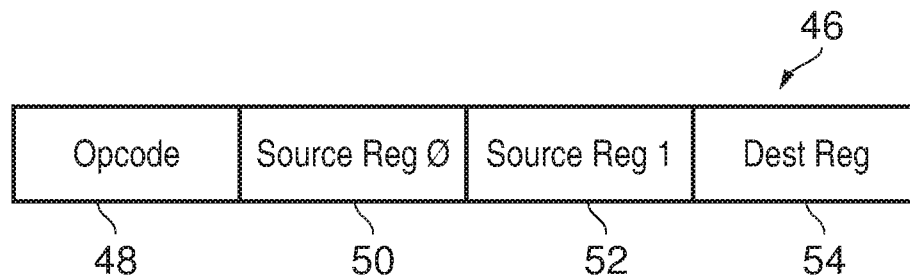


FIG. 2

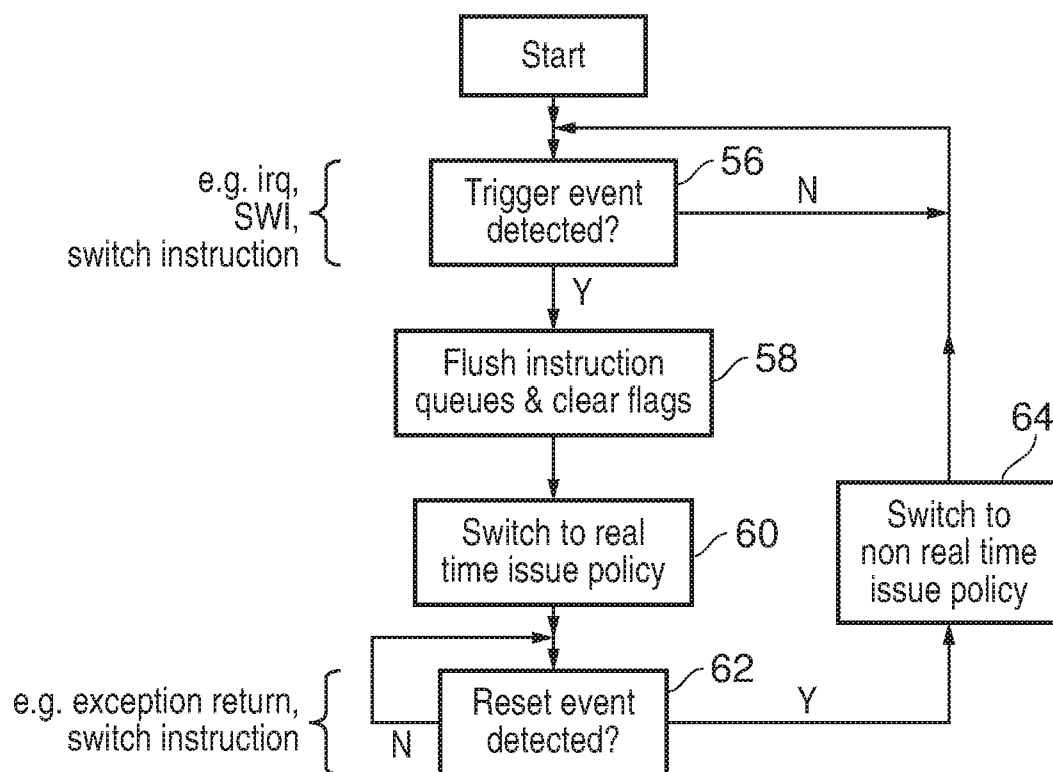


FIG. 3

3/5

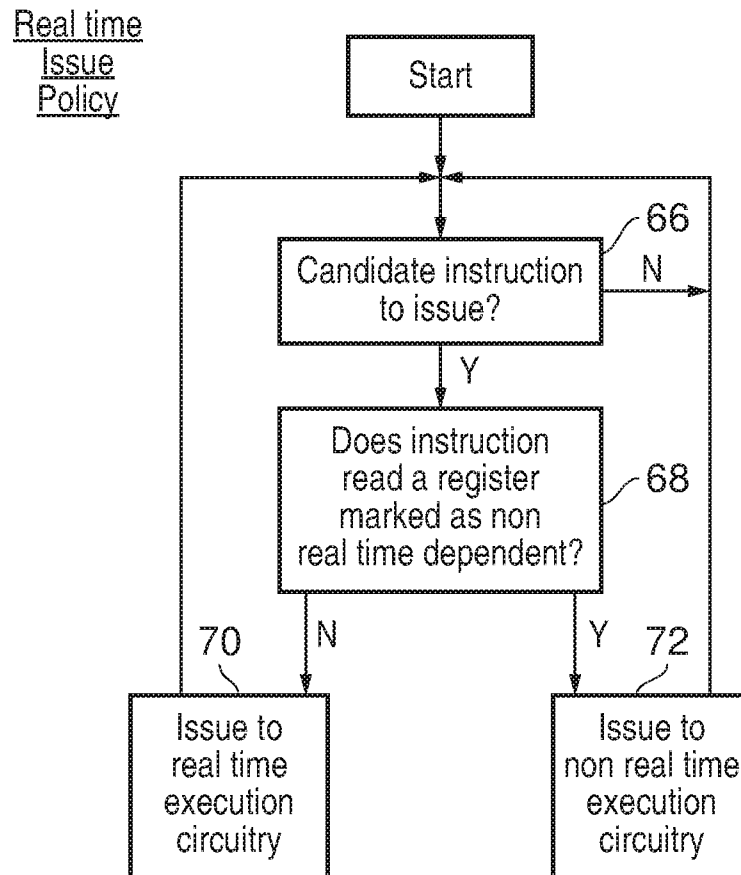


FIG. 4

4/5

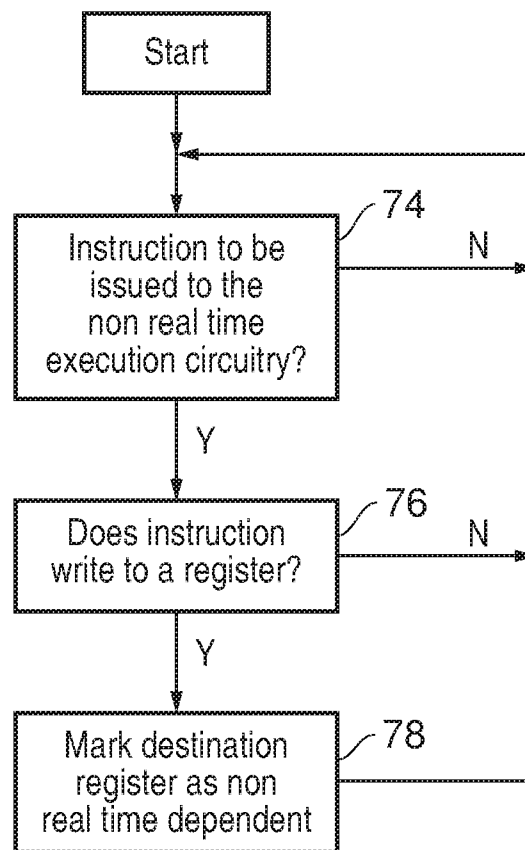


FIG. 5

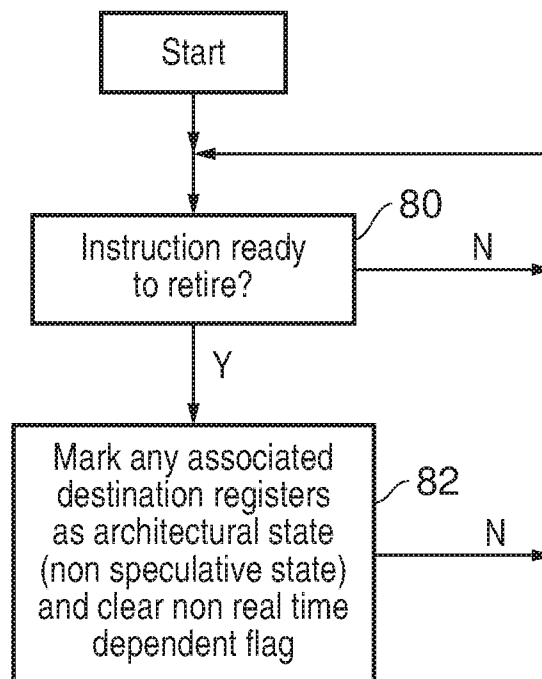


FIG. 6

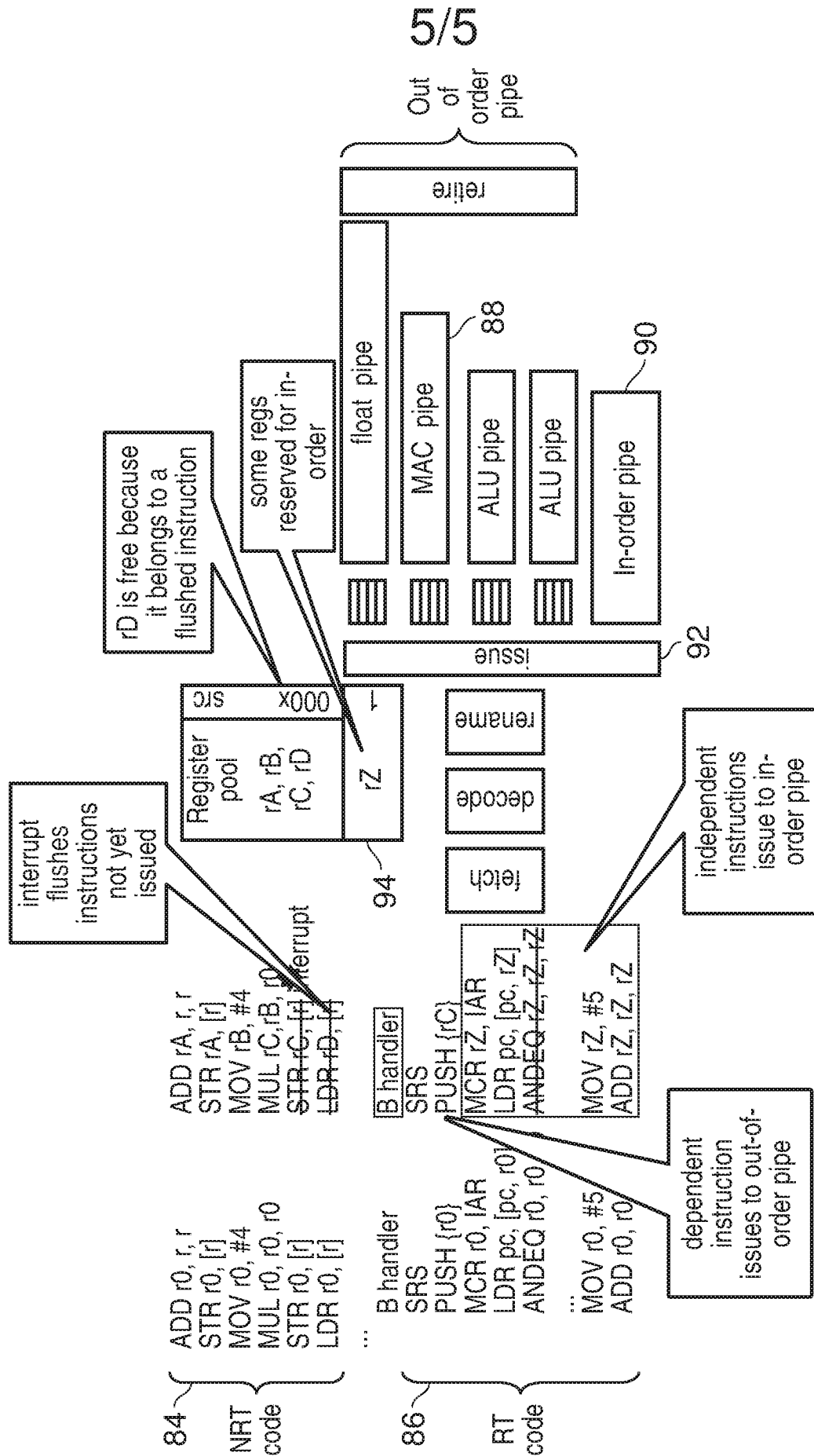


FIG. 7

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2016/051001

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/30 G06F9/38
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G05B G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	GB 2 527 927 A (UNIV MICHIGAN [US]) 6 January 2016 (2016-01-06) the whole document	1,24,25
X	----- US 2008/250226 A1 (EICKEMEYER RICHARD JAMES [US] ET AL) 9 October 2008 (2008-10-09) paragraph [0007] paragraph [0047] - paragraph [0048]	1,24,25
X	----- US 2014/281402 A1 (COMPARAN MIGUEL [US] ET AL) 18 September 2014 (2014-09-18) the whole document ----- -/-	1,24,25



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

10 June 2016

Date of mailing of the international search report

20/06/2016

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Klocke, Lynn

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2016/051001

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2015/006856 A1 (RASH WILLIAM C [US] ET AL) 1 January 2015 (2015-01-01) paragraph [0081] paragraph [0086] paragraph [0075] -----	1,24,25
A	US 2006/212682 A1 (BAXTER JEFFERY J [US] ET AL) 21 September 2006 (2006-09-21) paragraph [0022] - paragraph [0023] -----	1,24,25

INTERNATIONAL SEARCH REPORT

International application No.
PCT/GB2016/051001

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☒ Claims Nos.: 2-23(completely); 1, 24, 25(partially)
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
see FURTHER INFORMATION sheet PCT/ISA/210
3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☐ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

Continuation of Box II.2

Claims Nos.: 2-23(completely); 1, 24, 25(partially)

opinion as to novelty, inventive step and industrial applicability for the claims 1-25.

Due to the extent of lack of compliance with Articles 5 and 6 PCT, it is not possible to make a complete search of the claims as presently worded. The subject-matter of claims 1,24,25, has been searched, as far as this can be understood, claims 2-23 have not been searched. The search is has been carried out in the light of the description, taking the lack of compliance with Articles 5,6 PCT into account. The subject-matter of claim 1 which has been searched is as follows (interpretations from the description in square parentheses): Apparatus for processing data comprising:

issue circuitry to issue
processing operations [instructions] for execution;
non real time
execution circuitry [out-of-order pipelines 24, 26, 28, 30] to execute
processing operations [instructions] issued to said non real time
execution circuitry [out-of-order pipelines] by said issue circuitry;
real time execution circuitry [in-order pipeline 32] to execute
processing operations issued to said real time execution circuitry
[in-order pipeline 32] by said issue circuitry;
one or more registers
[in a register allocation table of a rename stage] to store data values,
respective registers of said one or more registers having associated
indicators to indicate for a given register when said given register is a
register allocated to store a data value written by an uncompleted
processing operation [instruction] issued to said non real time execution
circuitry [out-of-order pipelines]; and
issue policy control circuitry
to respond to a trigger event to enter a real time [in-order execution]
mode to control said issue circuitry to issue a candidate processing
operation [instruction] to one of said non real time [out-of-order]
execution circuitry and said real time [in-order] execution circuitry in
dependence upon whether that said candidate processing operation
[instruction] reads a register [having its associated indicator set].
The subject-matter of the other independent claims 24,25 has been
searched correspondingly.

The further subject-matter of the dependent
claims has not been searched.

The applicant's attention is drawn to the fact that claims relating to inventions in respect of which no international search report has been established need not be the subject of an international preliminary examination (Rule 66.1(e) PCT). The applicant is advised that the EPO policy when acting as an International Preliminary Examining Authority is normally not to carry out a preliminary examination on matter which has not been searched. This is the case irrespective of whether or not the claims are amended following receipt of the search report or during any Chapter II procedure. If the application proceeds into the regional phase before the EPO, the applicant is reminded that a search may be carried

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

out during examination before the EPO (see EPO Guidelines C-IV, 7.2), should the problems which led to the Article 17(2) declaration be overcome.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/GB2016/051001

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
GB 2527927	A	06-01-2016	CN 105242963 A	13-01-2016
			GB 2527927 A	06-01-2016
			KR 20160004920 A	13-01-2016
			US 2016004534 A1	07-01-2016

US 2008250226	A1	09-10-2008	CN 101281460 A	08-10-2008
			US 2008250226 A1	09-10-2008

US 2014281402	A1	18-09-2014	NONE	

US 2015006856	A1	01-01-2015	CN 105453030 A	30-03-2016
			EP 3014422 A1	04-05-2016
			KR 20150139931 A	14-12-2015
			TW 201523439 A	16-06-2015
			US 2015006856 A1	01-01-2015
			WO 2014209747 A1	31-12-2014

US 2006212682	A1	21-09-2006	US 2004059898 A1	25-03-2004
			US 2006190705 A1	24-08-2006
			US 2006190706 A1	24-08-2006
			US 2006212682 A1	21-09-2006
